ALISON: Fast and Effective Stylometric Authorship Obfuscation

Eric Xing[1], **Saranya Venkatraman**[2], **Thai Le**[3], **Dongwon Lee**[2]

[1]McKelvey School of Engineering, Washington University in St. Louis, MO, USA
[2]College of Information Sciences and Technology, The Pennsylvania State University, PA, USA
[3]School of Engineering, University of Mississippi, MS, USA
e.xing@wustl.edu, saranyav@psu.edu, thaile@olemiss.edu, dongwon@psu.edu

Abstract

Authorship Attribution (AA) and *Authorship Obfuscation* (AO) are two competing tasks of increasing importance in privacy research. Modern AA leverages an author's consistent writing style to match a text to its author using an AA classifier. AO is the corresponding adversarial task, aiming to modify a text in such a way that its semantics are preserved, yet an AA model cannot correctly infer its authorship. To address privacy concerns raised by state-of-the-art (SOTA) AA methods, new AO methods have been proposed but remain largely impractical to use due to their prohibitively slow training and obfuscation speed, often taking hours. To this challenge, we propose a practical AO method, ALISON, that (1) dramatically reduces training/obfuscation time, demonstrating more than 10x faster obfuscation than SOTA AO methods, (2) achieves better obfuscation success through attacking three transformer-based AA methods on two benchmark datasets, typically performing 15% better than competing methods, (3) does not require direct signals from a target AA classifier during obfuscation, and (4) utilizes unique stylometric features, allowing sound model interpretation for explainable obfuscation. We also demonstrate that ALISON can effectively prevent four SOTA AA methods from accurately determining the authorship of ChatGPT-generated texts, all while minimally changing the original text semantics. To ensure the reproducibility of our findings, our code and data are available at: https://github.com/EricX003/ALISON.

Introduction

Writing styles are often consistent among texts written by the same author. However, the writing styles of different authors can be very dissimilar. Therefore, the authorship identity of an anonymous piece of writing can still be revealed by analyzing its writing style and matching it to a pool of known authorship markers, a task known as **Authorship Attribution** (AA). In a machine learning context, authorship markers are predictive signals that can distinguish one author's writing style from the others. Such signals are often called stylometric features. Multiple types of stylometric features, including lexical features (e.g., structure of words and frequency of different character sequences), syntactic features (e.g., part-of-speech distributions and occurrences of functional words and punctuation), and content features

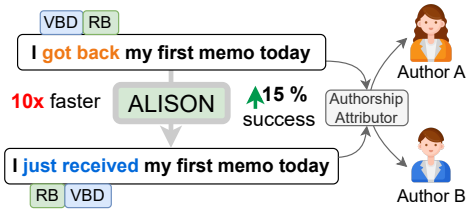

Figure 1: An example of ALISON successfully obfuscating a text by changing its style while preserving semantics.

(e.g., semantics of words and phrases in the text) are engineered to allow a machine learning model to match a text to an authorship label. These engineered features, such as Writeprints (Abbasi and Chen 2008), often include not one but several interpretable signals such as word and character bigrams, word length distributions, or special character frequencies to improve the classification accuracy.

However, recent AA techniques (Fabien et al. 2020; Devlin et al. 2019) utilize complex transformer models– e.g., BERT (Jin et al. 2020), RoBERTa (Liu et al. 2019), BertAA (Fabien et al. 2020), to automatically learn useful features for AA from raw text. This removes the need to rely on explicitly engineered stylometric features. While these models are more computationally expensive to train and notorious for their lack of interpretability, they significantly outperform traditional AA classifiers (Fabien et al. 2020).

As AA techniques become more accurate and efficient, they are more likely to be exploited by malicious actors to detect authorship identities behind anonymous texts. This is severely detrimental to a number of groups, especially NGO activists, whistleblowers, and journalists. As current SOTA transformer-based AA models are sufficiently powerful, it becomes important to develop methods that reduce the risk of an anonymous text's true authorship being exposed. Therefore, in this work, we study the opposite task of AA, known as **Authorship Obfuscation** (AO), which aims to thwart authorship attribution classifiers by making a few changes to the input text in a systematic way. Successful AO will fool the target model into making an incorrect attribution out of a pool of candidates. Because AA techniques generally degrade in performance as the number of authors

becomes large (i.e. > 100), and an adversary can generally narrow the pool of authors down to a small, finite set, we do not consider authorship obfuscation in the open-world setting. Figure 1 shows an example of a successful authorship obfuscation against a BERT-based (Devlin et al. 2019) authorship attributor.

There are three important properties that we desire in a "practical" AO approach: (1) ability to operate without significant knowledge of the adversary, (2) fast running time for long-form texts ($<$ 1 second, not minutes or hours), and (3) intuitive interpretability for a trustworthy obfuscation process. Unfortunately, SOTA AO methods, do not satisfy these properties at all, often requiring long running time to obfuscate text in a black-box fashion while making numerous calls to the attacked model. Such methods are impractical because a black-box understanding of the model to be attacked is often impossible to obtain, prohibitive running times diminish the productivity of an author seeking anonymity, and a lack of interpretability during obfuscation prevents current methods from being trustworthy.

To address the aforementioned limitations of current AO methods, we propose a novel stylometry-grounded novel obfuscation method, ALISON: (***Fast Stylometric Authorship Obfuscation***), which overcomes these challenges as follows:

- ALISON significantly reduces the obfuscation runtime by over 10x while also achieving better semantic preservation during obfuscation.
- ALISON consistently outperforms competing approaches by around 15% in obfuscation success rate.
- ALISON is also able to provide explanations for its obfuscation results with interpretable stylometric features.

Background

We narrow the scope of our work to the blind AO setting where a textual adversarial attack against AA classifier has two main constraints: (1) the attacker cannot query the AA classifier, and (2) the attacker also does not have access to its architecture, training data, etc. These constraints make the AO task more challenging but also more practical than existing threat models often used in existing literature where a public API to the target AA classifier is assumed to be accessible. The following section describes existing work pertinent to this AO setting.

Mutant-X (Mahmood et al. 2019) is an automated obfuscation method that utilizes genetic algorithms to iteratively make single-word substitutions by examining the confidence degradation gleaned from a black-box understanding of the attacked model. While this is a black-box attack method, we repurposed it as a blind attack as described in transferability studies associated with its original paper (Mahmood et al. 2019). Avengers Ensemble (Haroon et al. 2021) attempts to improve upon Mutant-X and decrease reliance on black-box knowledge of the target classifier by utilizing an ensemble-based internal classifier to improve the transferability of the method to a variety of adversaries, which boosts its performance in the blind attack setting. We will refer to this method as Avengers for the rest of the paper.

Other popular greedy-based black-box methods in the NLP adversarial literature, such as TextFooler (Jin et al. 2020) and BERT-Attack (Li et al. 2020), often have a high degree of dependence on the accessibility to the target AA classifier they attack. These methods make queries to the victim model per token in order to obtain a logit-based ranking of word importance. Then, top tokens may be replaced with close neighbors in precomputed embedding spaces (Jin et al. 2020) or by leveraging token representations of large language models (Li et al. 2020). However, these methods often demonstrate a sharp decline in performance once the attacks are transferred to different target classifiers (Jin et al. 2020). Additionally, such methods generally lack interpretability, as model explanations are based solely on the black-box model that is being attacked instead of revealing identifying linguistic patterns.

Lastly, large generative language models, such as ChatGPT (Ouyang et al. 2022), have demonstrated impressive paraphrasing capability which may be suitable for AO applications. A user may obtain a stylometrically different but semantically consistent text by prepending a fixed paraphrasing prompt to query a language model.

Problem Formulation

Given a text corpus $\mathcal{X}$, we define an AA classifier f trained on $\mathcal{X}$, such that for arbitrary text $x \in \mathcal{X}$, $f(x)$ attributes the authorship of x. Given $\mathcal{T}$ is a set of texts to obfuscate, our objective is to thwart f for any text $t \in \mathcal{T}$ by transforming t into t' such that $f(t) \neq f(t')$. We assume that $\mathcal{X}$ and $\mathcal{T}$ share the same pool of potential authors and are in a similar domain–e.g., news articles, blog posts– but do not contain any identical texts.

Moreover, we also assume no access to $\mathcal{X}$ by the adversary. However, they do have access to another non-overlapping corpus $\mathcal{X}^*$ with a similar size containing the same pool of authors and domain with $\mathcal{X}$. Such assumption is reasonable in practice, especially when online social networks have made it very convenient for anyone to access text content generated by millions of people worldwide. To evaluate our approach in this setting, we split each publicly available text classification corpus into three disjoint sets, $\mathcal{X}$, $\mathcal{X}^*$, and $\mathcal{T}$ stratified by unique authorship labels.

Proposed Method

Figure 2 illustrates ALISON's overall obfuscation pipeline. ALISON is designed to reduce computational complexity while advancing obfuscation success and semantic preservation during obfuscation. To do this, we employ three overarching strategies. First, we train an internal, lightweight AA classifier *once* that uses intuitive linguistic properties of part-of-speech (POS) sequences to guide the obfuscation process. Second, we aim to obfuscate a phrase of multiple words at a time instead of perturbing token by token. Third, we leverage an advanced pre-trained language model (PLM) to generate the replacement token sequence that best fits the sentence context and semantics without making queries to an embedding space.

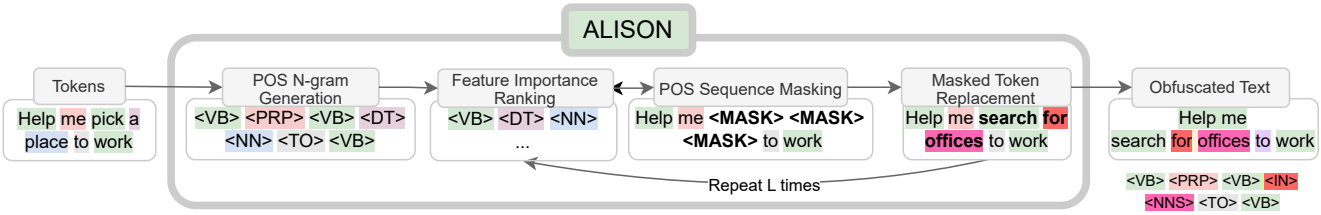

Figure 2: ALISON: Our proposed obfuscation pipeline.

One-Time Stylistic Internal AA Classifier Training

Because blind attacks on AA models often rely on an internal approximation of an arbitrary adversarial classifier to choose candidate words or phrases to be replaced, tuning the internal classifier for maximal transferability to other target classifiers is integral to producing high obfuscation success rate (Haroon et al. 2021). Therefore, we augment the traditional internal classifier feature space of character n-grams with POS n-grams, features we believe to be more heavily rooted in true style. We hypothesize that while writing style encompasses word and character frequencies, more generally, writing style also encompasses frequencies of individual POS tags and their collocations. Intuitively, POS and sequences of several POS tags capture writing style because they do not describe the content of the text but rather how the ideas in the text are synthesized. Generally, an author's texts should contain similar POS sequence patterns, as they represent common textual structures used to synthesize different ideas.

Feature Extraction. We first extract the POS tags of all texts in the corpus $\mathcal{X}^*$. Next, we extract character and POS tag n-grams of various lengths as features for training the internal classifier. Figure 3 demonstrates the procedure of extracting POS n-grams from a sample sentence with $n \leftarrow 3$.

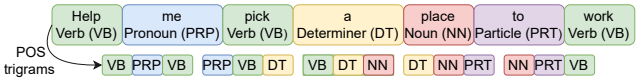

Figure 3: An example of extracting POS trigrams.

An n-gram is a contiguous sequence of n linguistic units (e.g., characters, words, POS tags) within a text. Given a set of sequence lengths V, for each length $l \in V$, we extract all character and POS level l-grams over the entire training corpus and collect the L most frequent character and POS l-grams. The normalized frequencies of these L most frequent character and POS l-grams for each length $l \in V$ are concatenated to form the stylistic representations of the text.

Internal Classifier Training. The resulting vector representations are then used to train a fully connected neural network (NN) model on the authorship attribution task. We opt for a simple NN due to its computational efficiency without much compromise on generalization. To utilize this model for prioritizing which phases or words in a sentence to perturb first, we then extract a list of features, ranked by importance, for $\forall t \in T$ using Integrated Gradients (Sundararajan, Taly, and Yan 2017), a model interpretability algorithm that assigns an importance score to each input feature by approximating the integral of the gradients with respect to the input.

We also multiply each extracted importance by the term $c^{length(feature)}$ for each feature's attribution, where c is a constant. During experimentation, we empirically observed that shorter POS n-grams were more abundant at the beginning of the attribution-ranked n-gram lists. We believe that this behavior is because of the necessarily lesser frequency of an arbitrarily longer POS n-gram in typical texts, as each longer n-gram occurrence necessarily is an occurrence of all contiguous substrings of the n-gram, i.e., shorter n-grams. Therefore, we introduced this scaling constant to artificially inflate the importance of longer POS n-grams to compensate for this behavior.

Replacement Phrase Generation via Masked PLM

To perform obfuscation, we must be able to generate replacement phrases using existing phrases as prompts. To do this, we leverage the masked language modeling approach used by Devlin et al. (2019). More specifically, given a sentence and the desired word tokens to be replaced, we mask the tokens to be replaced and use this modified text as input for a BERT model under a masked token prediction task. The top prediction for each masked token is used as the word's replacement. By using a SOTA language model, we aim to minimize the degree of information loss, as the language model will be able to infer much of the contents of the phrase through context but may scramble POS sequences, which hides authorship. This token-sequence masking procedure lies at the core of ALISON's speed-up, allowing a single PLM forward pass to perturb multiple tokens.

Text Obfuscation Process: One N-Gram at a Time

To obfuscate each $t \in T$, we first extract the POS tags and n-gram features for t, which are used to compute importance values as described previously. Then, we iterate through the ranked feature list in descending order of importance, omitting character n-gram features (only considering POS n-gram features) and pick the top L features. We omit character n-grams because important character n-grams are generally functional words or involve punctuation, which would negatively impact fluency upon perturbation.

Next, we attempt to match each of the top L POS n-grams to the POS n-gram profile of t. For each n-gram match

found, we update t through the phrase generation procedure as described previously. Lastly, we mark this phrase as changed so that it cannot be changed in subsequent steps as to prevent any specific section of text from deviating significantly from the original. Obfuscation is complete once all matches for the top L POS n-grams are processed.

One unique property of ALISON is that it will modify the text even if the internal classifier believes it will be classified incorrectly. This property is desirable because ALISON will uniformly obfuscate all texts, likely decreasing adversarial classifier confidence even if a complete obfuscation is unsuccessful. This differs from logit query-based methods because they do not attempt to perform *any* obfuscation if their internal classifier's prediction does not match the ground truth, leading to a large proportion of $t \in \mathcal{T}$ being completely unedited and therefore vulnerable.

Experimental Setup

Datasets. We use *TuringBench* (Uchendu et al. 2021) to evaluate ALISON on machine-generated texts. TuringBench is a collection of 160K human and machine-generated texts across 20 authors, 19 of which are neural text generation models, and one of whom is human. We also use the *Blog Authorship Corpus* (Schler et al. 2006) to evaluate ALISON on human-written texts. The dataset consists of the aggregated blog posts of 19,320 bloggers gathered from blogger.com, of which we select only the blogs from the top-10 most frequent authors. Both datasets are publicly available. We report all AO results on the test set.

Target Classifiers. We use three SOTA transformer-based models as target AA classifiers to attack: BERT (Devlin et al. 2019), DistilBERT (Sanh et al. 2019), and RoBERTa (Liu et al. 2019). These adversarial classifiers were trained on the *1st disjoint half* of the training and validation sets. They achieved around 80% testing accuracy on on TuringBench, while demonstrating varying performance on the Blog Authorship Corpus, ranging from approximately 85% (DistilBERT) to 95% (RoBERTa) testing accuracy.

Obfuscation Baselines and Internal Classifier Training. We utilize TextFooler, Mutant-X, Avengers, BERT-Attack, and ChatGPT as baselines to compare against our proposed AO framework ALISON. Except for ChatGPT, these methods all maintain an internal classifier for reference during obfuscation. While many of these are black-box attack methods, we repurposed them for the blind attack setting using the internal classifier specifications given in transferability studies instead of giving them access to our SOTA target models. Our neural-network-based n-gram classifier is trained on the disjoint 2nd half of the training and validation data that was not used to train our SOTA target models using $V = \{1, 2, 3, 4\}$. Internal classifiers for Mutant-X and Avengers were trained as outlined by their papers (Haroon et al. 2021; Mahmood et al. 2019) on the same data as our internal classifier. TextFooler was trained with both word-based CNN (wordCNN) (Kim 2014) and word-based LSTM (wordLSTM) internal classifiers as specified in their public implementation. We additionally tested TextFooler using our n-gram-based NN model (denoted as TextFooler-POS) to provide a fair comparison and illustrate the effectiveness of our stylometry-grounded approach. BERT-Attack was trained using standard BERT (Devlin et al. 2019). ChatGPT-based obfuscation was performed by pretending a fixed paraphrasing prompt to each text and obtaining the returned machine response.

Evaluation Metrics

- **Obfuscation Success.** The most intuitive measure of obfuscation success is measuring the target AA model's accuracy. Because there is a potential for the label distribution to become skewed during the removal of misclassified samples, we also measure F1-Score, a more robust metric in such a setting. To analyze the obfuscation success, we also monitor the reduction in target model accuracy between the original and obfuscated texts. Because we only retain correctly classified samples for obfuscation, the baseline accuracy and F1-Score are 1.00. A smaller post-obfuscation accuracy and F1-Score indicates a more successful attack, and therefore greater obfuscation success.
- **Running Time.** First, we recorded the running time of each algorithm, as an obfuscation method that requires a prohibitive amount of resources or computation time may not be scalable to real world AO scenarios. We split this time measurement into two phases, the time associated with one-time training of internal classifiers, and the time associated with the average inference time of the retained samples.
- **Semantic Preservation.** We also measure metrics of semantic preservation or semantic similarity between the original and obfuscated texts. Metrics indicating higher semantic preservation are favorable, as they indicate that there was a limited degree of information loss and that the perturbations to the text would not significantly impair a reader's understanding of the original text. These metrics include (1) *METEOR Score:* METEOR score is a standard for measuring the similarity between two texts in a natural language setting. It is grounded in the measure of alignments of word unigrams among texts; (2) *USE Cosine Similarity:* The Universal Sentence Encoder (USE) (Cer et al. 2018) is a text embedding model that is frequently adopted to accurately capture the semantics of a sentence. We utilize *cosine-similarity* to determine the degree of similarity between generated embeddings; (3) *BERTScore:* BERTScore (Zhang et al. 2020) is another metric of semantic similarity that utilizes BERT's pretrained contextual embeddings. BERTScore is calculated by maximizing pairwise embedding similarities for the tokens of an original and its obfuscated text. All scores lie in $[0, 1]$, and higher scores denote greater semantic similarity.
- **Fluency.** Lastly, we measure the perplexity of obfuscated texts to ensure that the obfuscation process does not diminish the human readability of obfuscated texts. The perplexity is calculated as the negative log-likelihood of LLaMA2-7B (Touvron et al. 2023) over obfuscated texts.

Results

Obfuscation Success. The experimental results on both datasets from our main obfuscation experiment are summarized by Table 1. In the table, we denote the metric indicat-

Method	Obfuscation Success (Lower is Better)		Semantic Preservation (Higher is Better)		
	Accuracy↓	F1-Score↓	METEOR↑	USE Cosine Similarity↑	BERTScore↑
TuringBench					
BERT					
Mutant-X	0.8987	0.8798	0.8381	0.9159	0.9366
Avengers	0.8354	0.8334	0.8333	0.9030	0.9320
TextFooler-wordCNN	0.7089	0.6797	0.8667	0.9614	0.9386
TextFooler-wordLSTM	0.7342	0.6935	**0.8813**	0.9671	0.9430
TextFooler-POS	0.7595	0.7011	0.8650	0.9635	0.9382
BERT-Attack	0.9114	0.9179	0.8388	0.8701	0.9526
ChatGPT	0.7089	0.6566	0.8373	0.9113	0.9490
ALISON	**0.6962 (-1.79%)**	**0.6065 (-7.63%)**	0.8505 (-3.49%)	**0.9682 (0.11%)**	**0.9583 (0.60%)**
DistilBERT					
Mutant-X	0.9494	0.9464	0.8450	0.9192	0.9406
Avengers	0.9113	0.8515	0.8341	0.9048	0.9320
TextFooler-wordCNN	0.7848	0.7556	0.8641	0.9609	0.9413
TextFooler-wordLSTM	0.7722	0.7705	**0.8819**	0.9677	0.9447
TextFooler-POS	0.7972	0.7955	0.8675	0.9657	0.9391
BERT-Attack	0.8228	0.8172	0.8434	0.8737	0.9538
ChatGPT	0.7456	0.6474	0.8428	0.9142	0.9494
ALISON	**0.5823 (-21.90%)**	**0.4925 (-23.93%)**	0.8538 (-3.19%)	**0.9685 (0.08%)**	**0.9588 (0.52%)**
RoBERTa					
Mutant-X	0.9014	0.8527	0.8182	0.9062	0.9306
Avengers	0.8028	0.7393	0.8157	0.8967	0.9248
TextFooler-wordCNN	0.6901	0.6074	0.8621	0.9618	0.9386
TextFooler-wordLSTM	0.7606	0.6682	**0.8814**	0.9686	0.9446
TextFooler-POS	0.7606	0.6760	0.8623	0.9624	0.9402
BERT-Attack	0.8451	0.8412	0.8279	0.8603	0.9484
ChatGPT	0.7924	0.6569	0.8268	0.9057	0.9436
ALISON	**0.6620 (-4.07%)**	**0.5624 (-7.41%)**	0.8554 (-2.95%)	**0.9701 (0.15%)**	**0.9595 (1.17%)**
Blog Authorship Corpus					
BERT					
Mutant-X	0.9130	0.9180	0.8325	0.8514	0.9237
Avengers	0.9565	0.9528	0.8894	0.9028	0.9316
TextFooler-wordCNN	0.9348	0.9305	0.8854	0.9472	0.9356
TextFooler-wordLSTM	0.9565	0.9531	0.8811	0.9439	0.9382
TextFooler-POS	0.9348	0.9476	0.8838	0.9453	0.9321
BERT-Attack	0.9130	0.8914	**0.9007**	0.9221	0.9202
ChatGPT	0.9022	0.8908	0.6720	0.8827	0.9368
ALISON	**0.8804 (-2.42%)**	**0.7860 (-11.76%)**	0.8296 (-7.89%)	**0.9551 (0.83%)**	**0.9386 (0.04%)**
DistilBERT					
Mutant-X	0.9048	0.9128	0.8209	0.8497	0.9135
Avengers	0.9405	0.9435	0.8826	0.9044	0.9305
TextFooler-wordCNN	0.8810	0.8570	0.8839	0.9465	0.9356
TextFooler-wordLSTM	0.8810	0.8425	0.8786	0.9427	0.9382
TextFooler-POS	0.8810	0.8591	0.8832	0.9442	0.9349
BERT-Attack	0.9048	0.8784	**0.9026**	0.9245	0.9205
ChatGPT	0.9762	0.9712	0.6524	0.8820	0.9347
ALISON	**0.7738 (-12.17%)**	**0.7189 (-14.67%)**	0.8431 (-6.59%)	**0.9595 (1.37%)**	**0.9387 (0.05%)**
RoBERTa					
Mutant-X	0.9895	0.9886	0.8285	0.8514	0.9232
Avengers	1.00	1.00	0.8886	0.9033	0.9331
TextFooler-wordCNN	0.3579	0.3397	0.8872	0.9496	0.9370
TextFooler-wordLSTM	0.3684	0.3394	0.8832	0.9464	0.9382
TextFooler-POS	0.3369	0.3295	0.8654	0.9417	0.9339
BERT-Attack	0.9053	0.8737	**0.9018**	0.9239	0.9205
ChatGPT	0.5684	0.5939	0.6682	0.8844	0.9368
ALISON	**0.3053 (-9.38%)**	**0.2912 (-11.62%)**	0.8288 (-8.09%)	**0.9544 (0.51%)**	**0.9452 (0.75%)**

Table 1: Results from main obfuscation trials, $15 < L < 25$. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.

Method	One-Time Training	Inference
TuringBench		
Mutant-X	4 hrs	3 min
Avengers	6 hrs	5 min
TextFooler-wordCNN	2 hrs	8 sec
TextFooler-wordLSTM	2 hrs	7 sec
BERT-Attack	6 hrs	8 sec
ALISON	**12 min**	**0.8 sec**
Blog Authorship Corpus		
Mutant-X	8 min	10 min
Avengers	24 min	14 min
TextFooler-wordCNN	2 hrs	11 sec
TextFooler-wordLSTM	2 hrs	9 sec
BERT-Attack	6 hrs	9 sec
ALISON	**6 min**	**1.0 sec**

Table 2: Statistics of the one-time training runtime and the average inference time per one sample for all methods.

ing the most favorable attack in bold (the metric with the lowest magnitude for obfuscation success metrics, and the metric with the highest magnitude for semantic preservation metrics) across each adversarial trial. Additionally, for the rows containing results for ALISON, we show the percentage change of each metric from the method that was the highest performing, excluding ALISON. Therefore a lower percentage (higher degradation of adversarial accuracy / F1-Score) is more desirable for obfuscation success metrics, while a higher percentage (less semantic degradation) is favorable for semantic preservation metrics.

On TuringBench, we see that ALISON is consistently the best performer in terms of attack success. ALISON consistently degrades adversarial accuracy more than other methods, demonstrating improvement as high as 21.90%. Additionally, F1-Score even more pronounced degradation, with improvement as high as 23.93%.

On the Blog Authorship Corpus, results shown in Table 1 indicate that ALISON is consistently the best performer in terms of F1-Score and accuracy.

Ablation of Interpretability-Based Replacement. We observe that ALISON outperforms TextFooler-POS in all trials. This demonstrates the value of ALISON's sequence replacement schema and interpretability-centric approach when compared to traditional token-by-token perturbation methods.

Computational Complexity. Running time results are summarized by Table 2. The One-Time Training stage encompasses all operations associated with data feature extraction and one-time training, while Inference corresponds to per-text running time.

The results indicate that ALISON outperforms all baselines both in terms of one-time training and obfuscation runtime. ALISON's total time for both one-time training and obfuscation of 100 samples indicates at least a 10x speed-up on TuringBench and at least an 18x speed-up on the Blog Authorship Corpus. ALISON is additionally at least 10x faster on TuringBench and 20x faster on the Blog Authorship Corpus with respect to one-time training and at least 10x faster during obfuscation on both datasets.

Semantic Preservation. Across both datasets, ALISON consistently outperforms in semantic preservation when evaluated with USE cosine similarity, the most robust measure of semantic preservation we measured, and BERTScore. However, we observe that ALISON consistently performs the worst in terms of METEOR score on both datasets; however, we believe that this result can largely be attributed to the inherent flaws of the METEOR score, as it is generally less correlated with human judgments when compared to USE cosine similarity, which is a stronger standard for semantic similarity analysis. We demonstrate these limitations in the Appendix.

Fluency. Table 3 demonstrates that ALISON demonstrates the best perplexity across both datasets, indicating the highest readability across all AO methods.

Method	TuringBench	Blog
Mutant-X	65.12	29.55
Avengers	64.51	23.12
TextFooler-wordCNN	57.69	17.96
TextFooler-wordLSTM	52.89	19.28
TextFooler-POS	56.23	18.34
ALISON	**20.82**	**12.11**

Table 3: Perplexity of post-obfuscation texts measured using LLaMA2-7B (lower is better).

Discussion

Author Label Bias. First, we analyze the distribution of author frequencies before and after obfuscation to identify potential obfuscation bias towards an author or set of authors on both datasets. To do this, we calculate the normalized entropy of author labels over obfuscated samples.

Because of the varied attack successes of different methods, we do not consider the raw entropy values but instead, consider the proportion of the total label entropy each author contributes. The distribution of these label entropy proportions should be as uniform as possible so that each author label transforms in an unpredictable way. A non-uniform entropy distribution across authors indicates that the obfuscation of a small pool of authors' texts contributes significantly to the overall attack success. This indicates a bias during obfuscation in regard to the transformation of author labels, a bias that can potentially be exploited by the attacked model. If the post-obfuscation prediction label were predictable based on the pre-obfuscation prediction label, an adversary would be able to gain significant information about the authorship of a text based on the predicted author post-obfuscation. This bias is further not desirable since the authorship pool may vary from various obfuscation settings.

We present the individual author entropy contributions over all authors for all methods in Figure 4. It is visually apparent that the distribution of author entropy contributions is significantly more uniform for ALISON when compared to other methods. This indicates significantly less predictability and label bias during obfuscation when compared to other methods. There are very few labels with a small

Method	Obfuscation Success		Semantic Preservation		
	Accuracy ↓	F1-Score ↓	METEOR ↑	USE Cosine Similarity ↑	BERTScore ↑
GPT Output Detector - Base	0.5000	0.3670	0.6966	0.8754	0.8941
GPT Output Detector - Large	0.5682	0.3623	0.6948	0.8734	0.9017
GPTZero	0.6170	0.5323	0.6897	0.8717	0.8936
DetectGPT	0.5729	0.4984	0.7478	0.9030	0.9134

Table 4: ALISON's attack success and semantic preservation against four machine text detection models.

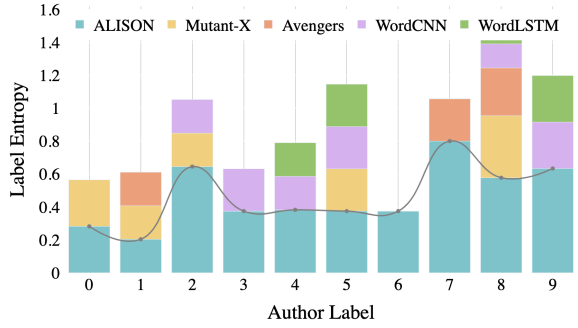

Figure 4: Distribution of author-wise contributions to label entropy post-obfuscation.

or nonexistent contribution to overall entropy, which are labels that could be trivially reverse-engineered by the targeted model, unlike the entropy distributions of other methods.

Interpretability. Because ALISON relies on explicitly determined criteria for obfuscation, it can explain obfuscation decisions using quantified token importances. Interpretability is generated by extracting the POS n-grams in a text and using Integrated Gradients to generate the importance of each POS n-gram, which is scaled as described previously. Top POS n-gram features may then be mapped to specific token sequences in the original text.

A Use Case: Obfuscating ChatGPT Texts

The impressive performance of ChatGPT (OpenAI 2023), a conversational language model, has led to its ubiquitous use in the workplace and classroom. Though ChatGPT can assist humans with everyday tasks, its potentially dishonest applications (e.g. construing ChatGPT's output as human-written text in academic settings) make the identification of ChatGPT-written texts an important problem with extensive commercial and academic study (Tian 2022; Mitchell et al. 2023; Solaiman et al. 2019; Wang, Le, and Lee 2023). The commercial value of ChatGPT detection further motivates an AO technique that is computationally efficient.

Problem Formulation. The real-world task of discriminating between ChatGPT and human-written texts is an increasingly relevant AA task that motivates the study of the corresponding AO task. We select four well-known machine-text generators, each demonstrating $> 95\%$ discrimination accuracy, to study under adversarial perturbation: GPTZero (Tian 2022), DetectGPT (Mitchell et al. 2023), and both the Base and Large GPT Output detectors (Solaiman et al. 2019) released by OpenAI.

Methodology. We used news article headlines from TuringBench to query the OpenAI Completions API. A single request was made for each unique headline, which consisted of a fixed generation prompt prepended to the headline. The corresponding human-written texts in the TuringBench corpus provided negative examples to introduce into the corpus, generating a set of evenly distributed negative and positive examples. The experimental setup described previously was then repeated.

Main Obfuscation Trial Result. Table 4 shows metrics of Obfuscation Success and Semantic Preservation against adversarial classifiers. ALISON demonstrates degradation of adversarial accuracy to *at most* 0.617 and adversarial F1-Score to *at most* 0.5323. In addition, ALISON consistently maintains a high degree of semantic similarity between original and obfuscation texts, maintaining *at least* 0.8717 USE Cosine Similarity and 0.8936 BERTScore. ChatGPT text detectors become negligibly useful at such adversarial performance, as the adversarial accuracy is close to the trivial accuracy of 0.50 in the binary classification setting.

Entropy Result. We observe an entropy of 0.56 associated with the human class and an entropy of 0.44 associated with the ChatGPT class. Because the distribution of authorship label entropy is not significantly skewed toward any class, ALISON does not demonstrate a significant degree of bias during the obfuscation process in transferring attributions from any specific class.

Conclusion

We have presented a new authorship obfuscation technique, ALISON, based on the replacement of revealing stylistic sequences. ALISON greedily replaces text sequences matching POS n-grams identified to be important by interpreting a lightweight neural network trained to perform authorship attribution using mixed n-grams. We use ALISON to attack three SOTA transformer-based attribution classifiers and demonstrate an improvement in obfuscation success and semantic preservation when compared to *seven* diverse baselines. We demonstrate that ALISON's intuitive and simple but effective nature demonstrates a drastic improvement in computational complexity compared to baseline methods. Parameter analysis, qualitative analysis of ALISON's obfuscated texts, limitations of METEOR score, etc. are presented in the Appendix.

Ethical Statement

While authorship obfuscation enables freedom of speech for various previously described individuals including whistleblowers and journalists, it also potentially permits malicious groups to stay hidden. We acknowledge such ethical concerns but stress the need to study and design systems that can protect and enhance the freedom of speech of the public.

Acknowledgements

This work was supported in part by NSF awards #1820609, #1950491, and #2131144.

References

Abbasi, A.; and Chen, H.-c. 2008. Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace. *ACM Transactions on Information Systems*, 26: 1–29.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Strope, B.; and Kurzweil, R. 2018. Universal Sentence Encoder for English. In *Conference on Empirical Methods in Natural Language Processing*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 4171–4186.

Fabien, M.; Villatoro-Tello, E.; Motlicek, P.; and Parida, S. 2020. BertAA : BERT fine-tuning for Authorship Attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, 127–137. Indian Institute of Technology Patna, Patna, India: NLP Association of India (NLPAI).

Haroon, M.; Zaffar, M. F.; Srinivasan, P.; and Shafiq, Z. 2021. Avengers Ensemble! Improving Transferability of Authorship Obfuscation. *ArXiv*, abs/2109.07028.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Li, L.; Ma, R.; Guo, Q.; Xue, X.; and Qiu, X. 2020. BERT-ATTACK: Adversarial Attack Against BERT Using BERT. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 6193–6202.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Mahmood, A.; Ahmad, F.; Shafiq, Z.; Srinivasan, P.; and Zaffar, F. 2019. A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X. *Proceedings on Privacy Enhancing Technologies*, 2019: 54–71.

Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C. D.; and Finn, C. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. arXiv:2301.11305.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schler, J.; Koppel, M.; Argamon, S.; and Pennebaker, J. W. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, 199–205.

Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J. W.; Kreps, S.; McCain, M.; Newhouse, A.; Blazakis, J.; McGuffie, K.; and Wang, J. 2019. Release Strategies and the Social Impacts of Language Models. arXiv:1908.09203.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*.

Tian, E. 2022. GPTZero. https://gptzero.me/. Accessed: 2023.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Uchendu, A.; Ma, Z.; Le, T.; Zhang, R.; and Lee, D. 2021. TuringBench: A Benchmark Environment for Turing Test in the Age of Neural Text Generation. In *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

Wang, Z.; Le, T.; and Lee, D. 2023. UPTON: Preventing Authorship Leakage from Public Text Release via Data Poisoning. In *Findings of Conf. on Empirical Methods in Natural Language Processing (EMNLP-Findings)*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *Int'l Conf. on Learning Representations (ICRL)*.